MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2022/23

ECE3186 - EMBEDDED IOT SYSTEMS AND APPLICATIONS

(All sections / Groups)

10 JULY 2023
9:00 a.m. – 11:00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 6 pages including cover page and appendix with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the answer booklet.

Question 1

In a sausage production factory, the meat must be thoroughly blended before being sent to a pressurized boiler for cooking. In an Industry 4.0-based facility, the processed meat is passed down through a conveyor belt to be transferred to the packaging machine.

Due to the conveyor belt's high-speed movement, operators are unable to consistently monitor the food being transferred for extended periods. This can result in some instances of unevenly cooked meat or unnoticed metal debris in the blended product.

i. Propose an embedded system solution to solve the problem with minimum human intervention. [9 marks]

ii. Draw a block diagram to show the functional units and the relationship in the embedded system that you have proposed. [6 marks]

iii. Elaborate in your own words, on the characteristics that your proposed system should possess. [5 marks]

iv. Explain in your own words, how the technology of internet connectivity can be employed to assist the microcontroller in food safety monitoring system. [5 marks]

Continued ...

Question 2

(a) A company aims to develop a product that allows for remote monitoring of a water dam in order to assess the water levels in the surrounding areas. Various sensor readings, such as rainfall, flow rate, and water levels at specific streams, must be taken into account. However, the area is not accessible via mobile cellular networks.

i. Evaluate the term LoRaWAN and NB-IoT in terms of utilization of technology, spectrum, protocol, and transmit current. [8 marks]

ii. Justify the most suitable wireless communication technology to be proposed for this embedded system. [4 marks]

(b) Analyze the impact of L1 cache on the performance of a given processor. Let τ be the access time for L1 caches and $x\tau$ for the main memory. To transfer a block from main memory to L1, the miss penalties are given as $(2*x)\tau$. Assume that the hit rate in L1 cache is $0.8x$. Use the non-zero last digit of your Student ID as the value of x. For example, if your Student ID is 1234567890, $x = 9$.

i. Calculate the average access time, as seen by the processor. [4 marks]

ii. Estimate the improvement in memory performance with the utilization of cache. [4 marks]

(c) Provide and elaborate THREE processor technologies in terms of their strengths and weakness. Select ONE processor technology which is suitable for used in a wearable medical assistant product.

[5 marks]

Continued ...

Question 3

(b) Briefly discuss TWO advantages and TWO disadvantages of the C programming language in comparison to the assembly programming language. [6 marks]

(b) As an engineering intern, you are tasked to develop an embedded system which controls a traffic light using an ESP32 development board. The system should have two LEDs representing the red light (connected to GPIO17, active HIGH) and green light (connected to GPIO18, active HIGH), and a button (connected to GPIO19, active LOW) to switch between the two lights. The program should initially turn on the red light and wait for a button press. When the button is momentarily pressed, the program should turn off the red light and turn on the green light. When the button is momentarily pressed again, the program should turn off the green light and turn on the red light. The program should repeat this sequence indefinitely. Refer to the appendix for a list of common ESP32 functions.

i. Write the PDL statements to show how the embedded program can be designed. [4 marks]

ii. Write an Arduino sketch for the system based on the PDL statement written above. [6 marks]

(c) As an intern at an embedded system solution provider, you are required to develop an Arduino sketch for an ESP32 development board:

i. Perform the following tasks in the Arduino sketch:
- Label GPIO4 and GPIO16 as BUTTON1 and LED1, respectively.
- Write a setup function to fulfil the following requirements:
  - Set up GPIO4 as an input pin with internal pull-up resistor.
  - Set up GPIO16 as an output pin.
  - Set up the serial communication at a baud rate of 9600 with the default data format of 8 data bits, no parity, and 1 stop bit (8N1).

[4 marks]

ii. Based on the setup function written in Q3(c)(i), write a loop function that continuously reads the state of BUTTON1 and controls LED1 accordingly, while also sending the button's status through serial communication. The function should fulfil the following requirements:
- When the button is pressed (the state of BUTTON1 is LOW), turn ON the LED1 (set LED1 to HIGH) and send "Button Pressed" through serial communication.
- When the button is not pressed (the state of BUTTON1 is HIGH), turn OFF the LED1 (set LED1 to LOW) and send "Button Not Pressed" through serial communication.
- Wait for one second before checking the button's status to repeat the operations.

Note: Refer to the appendix for a list of common ESP32 functions. [5 marks]

Continued ...

Question 4

(a) Draw a labeled diagram to illustrate the publish/subscribe communication model. Additionally, provide a brief explanation of its operations. [7 marks]

(b) Write a Python program to connect to a temperature sensor that publishes data to the topic "sensor/temperature" on a MQTT broker running on a Raspberry Pi with IP address 192.168.1.100 (port 1883). Store all messages received by appending them to a text file called "temperatures.txt". You can use the following Python code to import the relevant Python module for your program:

```
import paho.mqtt.client as paho
```

[6 marks]

(c) You have a CSV file called "mydata.csv" that contains a dataset with several columns of features with the target in the last (rightmost) column. Write a Python program that uses the Pandas library to load the dataset from the CSV file, separate the loaded dataset into features (X) and target (y), and print the shapes of X and y. The required Python modules have been loaded as shown in Program Listing Q4(c). [6 marks]

```
from pandas import read_csv
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split as split
```

Program Listing Q4(c)

(d) Analyse the Python program written in Q(4)(c) and extend it to train a k-Nearest Neighour (k-NN) model using 350 data samples from a total of 500 data samples, evaluate the trained model using the remaining 150 data samples, and finally print the accuracy of the trained model. Assume that all default settings are used in training the model [6 marks]

Continued ...

Appendix: Common ESP32 functions

Digital I/O Functions:

- **pinMode(pin, mode)**: Configures the specified pin to behave either as an input or an output.
- **digitalWrite(pin, value)**: Writes a HIGH or a LOW value to a digital pin.
- **digitalRead(pin)**: Reads the value from a specified digital pin, either HIGH or LOW.

Analog I/O Functions:

- **analogRead(pin)**: Reads the value from the specified analog pin.
- **analogWrite(pin, value)**: Writes an analog value (PWM wave) to a pin.

Time Functions:

- **millis()**: Returns the number of milliseconds since the board began running the current program.
- **delay(ms)**: Pauses the program for the specified number of milliseconds.
- **delayMicroseconds(us)**: Pauses the program for the specified number of microseconds.

Serial Communication Functions:

- **Serial.begin(baudrate)**: Sets the data rate in bits per second (baud) for serial data transmission.
- **Serial.print(data)**: Prints data to the serial port as human-readable ASCII text.
- **Serial.println(data)**: Prints data to the serial port as human-readable ASCII text, followed by a newline character.

Wi-Fi Functions:

- **WiFi.begin(ssid, password)**: Connects to a Wi-Fi network with the specified SSID and password.
- **WiFi.disconnect()**: Disconnects from the current Wi-Fi network.
- **WiFi.status()**: Returns the current Wi-Fi connection status.

PWM Functions:

- **ledcSetup(channel, frequency, resolution):** Configures a PWM channel with the specified frequency and resolution.
- **ledcAttachPin(pin, channel):** Attaches a pin to a PWM channel.
- **ledcWrite(channel, dutyCycle):** Writes a duty cycle to a PWM channel.

Task Management Functions:

- **xTaskCreate(taskFunction_t task, const char* const name, const uint32_t stackDepth, void* const parameters, UBaseType_t priority, TaskHandle_t* const handle):** Creates a new task and adds it to the list of tasks that are ready to run.
- **vTaskDelete(TaskHandle_t handle):** Deletes a task and removes it from the list of tasks that are ready to run.
- **vTaskDelay(pdMS_TO_TICKS(timeInMs)):** Suspends the execution of the current task for the specified number of milliseconds.

End of Paper